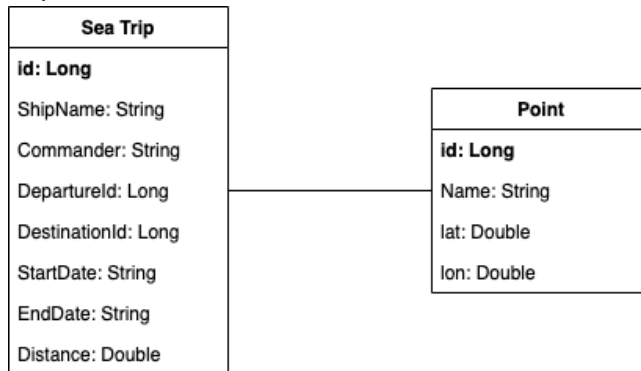


- Схема БД

SQL:



NOSQL:

```

{
  "_id": <ObjectId>,
  "shipName": "St. Elena",
  "commander": "David Porter Jr.",
  "departure": {
    "_id": <ObjectId>,
    "name": "New York",
    "lat": 134.2144,
    "lon": 95.313
  },
  "destination": {
    "_id": <ObjectId>,
    "name": "Los Angeles",
    "lat": 89.2144,
    "lon": 33.313
  },
  "startDate": <Timestamp>,
  "endDate": <Timestamp>,
  "distance": 9356.54
}
  
```

- Список сущностей

SQL:

Sea Trip – сущность морского путешествия, включает в себя следующие атрибуты:

Id – уникальный идентификатор

ShipName – Название судна

Commander – Имя командира

DepartureId – уникальный идентификатор начальной точки путешествия

DestanationId – уникальный идентификатор конечной точки путешествия

StartDate – дата начала путешествия в формате yyyyMMdd

EndDate – дата окончания путешествия в формате yyyyMMdd

Distance – расстояние пройденное судном за время путешествия

Point – точка на земном шаре (мб. город, порт, остров и тп.):

Id – уникальный идентификатор

Name – название точки

Lat – широта

Lon – долгота

NOSQL(MongoDB):

Каждое путешествие представляет собой отдельный документ.

Идентификатором документа является идентификатор путешествия. В документе есть поля “shipName”, “commander”, “startDate”, “endDate”, “distance”, “departure”, “destination”. Поля “departure”, “destination” являются вложенными документами, у которых есть поля “name”, “lat”, “lon”.

- Оценка объема информации

Пусть совокупность полей (“shipName”, “commander”, “startDate”, “endDate”, “distance”, “departure”, “destination”) занимает N памяти, M – число путешествий, вложенные поля “name”, “lat”, “lon” – занимают K памяти.

Тогда хранение данных:

В MongoDB будет занимать $M \cdot (N + 2 \cdot K) = M \cdot N + 2 \cdot M \cdot K$

В реляционной БД будет занимать $M \cdot N + 2 \cdot M \cdot K$

Тогда при условии того, что $N = (50 \text{ байт название судна} + 50 \text{ байт имя командира} + 8 \text{ байт дистанция} + 16 \text{ байта время начала и окончания})$, $K = (32 \text{ байта название} + 16 \text{ байт координаты})$, пусть число путешествий $M = 1200$.

Тогда хранение данных будет занимать:

В MongoDB – $1200 \cdot 124 + 2 \cdot 1200 \cdot 48 = 244\,800$ байт

В реляционной БД - 244 800 байт

Таким образом количество **чистой памяти**, занимаемое реляционной БД, не отличается от MongoDB.

Количество **памяти**, занимаемое MongoDB с учетом типов <Timestamp> (4 байта) и <ObjectId> (12 байт) = $1200 \cdot 116 + 2 \cdot 1200 \cdot 60 = 283\,200$ байт.

Количество **памяти**, занимаемое реляционной БД с учетом полей id (4 байта) = $1200 \cdot 136 + 2 \cdot 1200 \cdot 52 = 288\,000$ байт.

Таким образом MongoDB выигрывает реляционные БД **при учете всей информации хранимой в модели**.

Формула роста памяти для MongoDB при увеличении количества моделей будет иметь вид: $N \cdot (116 + 2 \cdot 60) = 236N$, для SQL модели: $240N$.

Следовательно, с увеличением количества записей реляционная БД будет занимать в 60/59 раз больше памяти.

- Запрос на поиск путешествия по id по всей коллекции путешествий (без учета индексов)

В MongoDB – $O(M)$

В SQL БД – $O(2 * (M * 2M)) = O(4M^2)$, т.к. необходимо сделать 2 раза операцию join на таблицу Point; размер таблицы Point - 2M, таблицы SeaTrips - M

- Запросы

Создание записи: "db.seaTrips.insertOne(
{

```
  _id:1,
  shipName: "St. Elena",
  commander: "David Porter Jr.",
  startDate: new Date(41242142),
  endDate: new Date(41242142),
  distance: 9356.54,
  departure: {
    "_id": 21,
    "name": "New York",
    "lat": 134.2144,
    "lon": 95.313
  },
  destination: {
    "_id": 22,
    "name": " Los Angeles",
    "lat": 184.2144,
    "lon": 75.313
  },
}
```

})"

Поиск записи: "db.seaTrips.find({_id:1})"

Удаление записи: "db.seaTrips.remove({_id:1})"